CLAIMS

1. Method of automatic generation, by means of a data processing system (40) associated with a program called a Configurator for creating a global simulation model of an architecture comprising models of integrated circuits under development that can constitute, with the help of the automatic Configurator, a machine or a part of a machine, and environment simulation models that make it possible to test and verify the circuit under development, a configuration definition file (FCONF) for components of the architecture, these components constituting fixed functional blocks for describing the functionalities of integrated circuits or parts of integrated circuits, the components being chosen by the user from a library of various component types and a library of environment components, in order to create the global model of the architecture corresponding to the functional specification defined in the configuration definition file (FCONF) and conforming to the specification of the architecture of the global model specified by an architecture description file (FDARCH), a method characterized in that it includes the following steps:

- the reading of the architecture description file (FDARCH) of the global model and the storage, in a component and connection rule table (TCRC), in a connection coherency rule table (TRCOH), and in a source file formatting table (TMFT), of information related to all of the possible configurations, each component obtaining a name (LABEL) that unambiguously identifies its position in the architecture, and a type from among several types (Active Components, Monitoring and Verification Blocks, Intermediate Blocks, System Blocks and Global Blocks),

- the instantiation of the components specified in the configuration definition file (FCONF) by the user-developer using a list of the components preset, designated by their names and types and including parameters or invoking procedures, the configuration definition file (FCONF) comprising a file from which to select the components and their types and optional additional indications concerning the type of interface and the server involved in the configuration to be generated by the Configurator, and the storage of the corresponding information in an instance connection table (TCINST),

- the topological connection of the instances and the storage of the corresponding information in an instance connection table (TCINST),

- the physical connection of the interface signals, at the level of each instance of the components, by applying regular expressions, stored in the component and connection rule table (TCRC), based on the names of the signals constituting a wiring table (TCAB),

- the use of the instance connection table (TCINST), the wiring table (TCAB) and the formatting table (TFMT) to automatically generate HDL-type (MGHDL) and HLL-type (MGHLL) source files of the global simulation model corresponding to the configuration specified by the configuration definition file (FCONF).

2. Method according to claim 1, wherein the Configurator system transmits to the HLL-type parts of each component information on:

- - the name (LABEL) of the component;
- the type of the instance (DUT, XACTOR, VERIFIER, MONITOR),
- the HDL path, i.e. the hierarchical name of the component in the description of the model.

3. Method according to claim 1 or 2, wherein the configuration definition file (FCONF) also includes a keyword (server<n>) indicating the name or number of the server in which a component is instantiated when the method is used in a multi-server system.

4. Method according to claim 4 wherein, in the case of a multi-server utilization, the configurator system executes the following steps:

- the division of the Configuration into several (HDL type and HLL type) parts, sorting the HDL-type components and the HLL objects according to the servers to which they belong,
- the generation of the HDL-type peripheral components used for sending and receiving signals between the parts of the configuration,
- the duplication of the Global Blocks by the Configurator system and the instantiation of the Global Blocks duplicated in each server,
- the generation of the HLL-type parts that serve as a communication medium between the servers.

5. Method according to claim 3 or 4, wherein the automatic connection between the components by the Configurator system includes several phases:

- a high-level topological phase for selecting the components and their respective positions,
- a wiring phase for creating the actual connection between the components, this phase generating as a result a wiring table (TCAB) that associates the signals connected to one another with the unique name of the wire that connects them,
- a phase for generating HDL-type and HLL-type source files.

6. Method according to claim 5, wherein the wiring phase is performed by the Configurator system in the following three steps:

a. the Global Blocks and the System Blocks are first connected to all of he components,

b. next come the connections of the signals between the other components,

c. after the wiring, an additional pass makes it possible to connect the remaining unconnected signals of each component to predetermined signals in order to produce a given stable state; the Configurator system then generates partial configurations comprising a subset of the architecture.

7. Method according to claim 6, wherein the predetermined signals are the signals of the System Block corresponding to the component.

8. Method according to any of claims 1 through 7, wherein the description file of the architecture (FDARCH) of the global model includes the simulation models of the Global Blocks and the System Blocks, these two types of components being connected to one another and handling environment signals.

9. Method according to claim 8, wherein the System Blocks are connected to the other components and supply them with the system signals that are specific to them

10. Method according to claim 9, wherein the data processing system (40) performs a conformity check of the connections, comparing the connection table of the real instances between blocks (TCINST) to the connection coherency rule table (TRCOH).

11. Method according to claim 10, wherein the data processing system (40) compares the physical connections between the components to the connection coherency rule table (TRCOH), in order to detect any incompatibilities between the ends of the connections between the components, and in such a case, it specifies, and adds into the instance connection table (TCINST), an adapter component (Intermediate Block) (101) inserted into the connection in question.

12. Method according to claim 11, wherein the configuration definition file (FCONF) includes information, specified by an attribute, concerning the utilization of adapter components (Intermediate Blocks) with the instances of the active Components, whose connections are compared to the instance connection table (TCINST), in order to detect any incompatibilities between the ends of the connections between the components, and in such a case it specifies, and adds into the instance connection table (TCINST), another adapter component (Intermediate Block) (102) inserted into the connection in question.

13. Method according to claim 12, wherein the data processing system (40) selects certain connections between the components of the connection coherency rule table (TRCOH) and specifies, and adds into the instance connection table (TCINST), additional connections constituting branches leading to respective additional models, which represent tools (probes) for monitoring the connections.

14. Method according to any of claims 1 through 13 wherein, in the source file generation nphase, the Configurator system generates the source files in HDL language (MGHDL) and in HLL language (MGHLL), based on the content of the component and connection rule table (TCRC), the coherency rule table (TRCOH), the source file formatting table (TMFT), the instance connection table (TCINST) and the wiring table (TCAB).

15. Method according to claim 14, wherein the data processing system (40) executes an operation through the Configurator system for each configuration variant, in order to obtain several simulation models corresponding to the same functional specification, but written in a description comprising various mixtures of languages of different levels (HDL, HLL).

16. Method according to any of claims 1 through 15, wherein the data processing system (40) generates the functional specification of the global simulation model in a computer format compatible with a high-level programming language, (HLL) and in a format compatible with a hardware description language (HDL).

17. Method according to either of claims 15 and 16, wherein the configuration definition file (FCONF) comprises, for each component, at least one part in HDL-type language, said part in HDL-type language providing an interface with other models.

18. Method according to claim 17, wherein the models that include a part in HLL-type language include interface adapters.

19. Method according to claim 18, wherein the Configurator system chooses each interface adapter model as a function of the connection coherency rule table (TRCOH).

20. Method according to claim 19, wherein the connections of the physical signals are specified by "Ports," each port being an arbitrary selection of the signals of the HDL-type interface of a component by means of regular expressions based on the names of these signals, and being constituted by regular expression/substitute expression pairs, these expressions being successively applied to the name of each signal of the HDL-type interface, and if the final substitution is identical for two signals, the latter are connected to one another, the connection being stored in the wiring table (TCAB).

21. Method according to claim 20 wherein, each interface adapter being shared among several models connected to the same port, only one of these models transmits signals through said port.

22. Data processing system (40) for automatically generating a global simulation model of a configuration of fixed functional blocks, mutually connected by interworking connections so as to constitute the global simulation model of an architecture comprising models of integrated circuits under development that can constitute a machine that conforms to the functional specification of a configuration, this system being characterized in that the data processing system (40) uses a Configurator program (PROGCONF) that includes means for creating a simulation of the wiring by applying stored regular expressions, and for using a configuration definition file (FCONF) in a high level language, a component and connection rule table (TCRC) describing the properties (type, HDL-type interfaces, ports, constructors of HLL class objects, etc.) of the software components for simulating the circuit, a connection coherency rule table (TRCOH) in a high level language (HLL), means for instantiating the elements resulting from the configuration definition file (FCONF), and an HLL code generator that combines the parameters of the components with the connection rules.

23. System according to claim 22, characterized in that there are at least five types of components: Active Components, Monitoring and Verification Blocks, Intermediate Blocks, System Blocks and Global Blocks.

24. System according to claim 23, characterized in that it is equipped to perform a conformity check of the connections by comparing the instance connection table (TCINST) with a table of coherency rules (TRCOH) for the physical connections between the models chosen from the blocks constituting the global model.

25. System according to claim 24, characterized in that it is designed to compare the connection table of the real instances (TCINST) between blocks to the connection coherency rule table (TRCOH), in order to detect any incompatibilities between the ends of the connections between blocks, and in such a case to specify, and add into the connection coherency rule table (TRCOH), a functional adapter block (Intermediate Block) (101) inserted into the connection in question.

26. System according to any of claims 22 through 25, characterized in that the component and connection rule table (TCRC), which includes the properties of the components, contains global parameters common to all of the component types and exists

in the form of a table distributed into one or more tables, which may or may not be associative, wherein the entries are names designating all of the possible models for the same component.

27. System according to claim 26, characterized in that the associative tables can contain the description, either in the form of parameter sets or in the form of references to procedures that generate the required values, the entries of these associative tables being names designating all of the possible models for the same component and forming a character string containing predetermined special identifiers, replaced by the values calculated by the Configurator system.

28. System according to claim 27, characterized in that at least three selectors indicate the instance to be used, the latter being transmitted as a parameter to a constructor of an HLL object:

- a first selector indicates the current instance ("item"),
- a second selector specifies the instance connected to the other end of the port,
- a third selector indicates the composite instance corresponding to the active Component containing the observation port.

29. System according to any of claims 22 through 28, characterized in that the Configurator system uses one or more connection coherency rule tables (TRCOH), which represent the rules for interconnecting the components and for inserting intermediate components, one or more component and connection rule tables (TCRC), which represent the system-level connection rules and the rules for generating connections between the signals, and one or more source file formatting tables (TFMT), which represent the rules for generating instances of HLL-type objects.

30. System according to any of claims 22 through 29, characterized in that the Configurator system uses:

- an HLL base class for uniquely identifying each object instantiated and for polling the configuration,
- means for generating and automatically instantiating System Blocks,

- means for tables associating the signals connected together under the unique name of the connecting wires

- means for using a formatting table to generate the source files in HDL and HLL.

31. System according to any of claims 22 through 30, characterized in that the operator functionally specifies the configuration in the highest level language as much as possible, and completes the functional specification with the components in the lowest level language.

32. System according to any of claims 22 through 31, characterized in that the following entries in the hash define the component Type (for example DUT (HDL model), XACTOR (transactor), MONITOR, VERIFIER or other types), and corresponds each Component Type to a hash, in turn composed of the following entries:

– a first entry *ReqModule* contains the name of the HDL module of the component and the name of the corresponding source file,

– a second entry *Connect* is the definition of the method for selecting the signals that are part of a Port, this description being composed of a set of entries indexed by the name of the Port; the configurator associates each Port name with a table of regular expressions and a pointer to a signal connection procedure that controls the application of these expressions to the names of the signals of the interface of the component.

33. System according to claim 32, characterized in that the generic structure of the active Components includes a Block containing the HDL description and a Block in HLL that provides the access paths to the HDL resources, and if necessary, a description of the block in HLL; the set of signals of the HDL block constitute the interface of the containing Block, formed by Ports, which are arbitrary logical selections of the signals of an interface, and by interface adapters, which are the software parts that handle, in each Port, the two-way communication between the parts in HLL and those in HDL, the interface adapters being chosen by the Configurator system.

34. System according to claim 33, characterized in that the Ports are specified in the form of regular expressions, which serve both to select the subsets of signals to be connected and to define the connection rules.

35. System according to any of claims 22 through 34, characterized in that the Configurator system generates so-called Transfer Components, which are inserted on each side of the cutoff to the servers, these components simply being wires for the inputs and registers for the outputs.